

02-25-00

A

UTILITY PATENT APPLICATION TRANSMITTAL (Large Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No.
2204/A12Total Pages in this Submission
32

TO THE ASSISTANT COMMISSIONER FOR PATENTS

Box Patent Application
Washington, D.C. 20231

Transmitted herewith for filing under 35 U.S.C. 111(a) and 37 C.F.R. 1.53(b) is a new utility patent application for invention entitled:

APPARATUS AND METHOD FOR BINDING A CONNECTION-ORIENTED CLIENT TO A
COMMUNICATION CHANNEL

and invented by:

Richard Crump

If a **CONTINUATION APPLICATION**, check appropriate box and supply the requisite information:
☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: _____

Which is a:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: _____

Which is a:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: _____

Enclosed are:

Application Elements

1. ☐ Filing fee as calculated and transmitted as described below
2. ☒ Specification having 20 pages and including the following:
 - a. ☒ Descriptive Title of the Invention
 - b. ☒ Cross References to Related Applications (if applicable)
 - c. ☐ Statement Regarding Federally-sponsored Research/Development (if applicable)
 - d. ☐ Reference to Microfiche Appendix (if applicable)
 - e. ☒ Background of the Invention
 - f. ☒ Brief Summary of the Invention
 - g. ☒ Brief Description of the Drawings (if drawings filed)
 - h. ☒ Detailed Description
 - i. ☒ Claim(s) as Classified Below
 - j. ☒ Abstract of the Disclosure

UTILITY PATENT APPLICATION TRANSMITTAL
(Large Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No.
2204/A12

Total Pages in this Submission
32

Application Elements (Continued)

3. ☒ Drawing(s) *(when necessary as prescribed by 35 USC 113)*
- a. ☐ Formal Number of Sheets _____
- b. ☒ Informal Number of Sheets 5
4. ☒ Oath or Declaration
- a. ☐ Newly executed *(original or copy)* ☒ Unexecuted
- b. ☐ Copy from a prior application (37 CFR 1.63(d)) *(for continuation/divisional application only)*
- c. ☒ With Power of Attorney ☐ Without Power of Attorney
- d. ☐ DELETION OF INVENTOR(S)
Signed statement attached deleting inventor(s) named in the prior application,
see 37 C.F.R. 1.63(d)(2) and 1.33(b).
5. ☐ Incorporation By Reference *(usable if Box 4b is checked)*
The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under
Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby
incorporated by reference therein.
6. ☐ Computer Program in Microfiche *(Appendix)*
7. ☐ Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all must be included)*
- a. ☐ Paper Copy
- b. ☐ Computer Readable Copy *(identical to computer copy)*
- c. ☐ Statement Verifying Identical Paper and Computer Readable Copy

Accompanying Application Parts

8. ☐ Assignment Papers *(cover sheet & document(s))*
9. ☐ 37 CFR 3.73(B) Statement *(when there is an assignee)*
10. ☐ English Translation Document *(if applicable)*
11. ☐ Information Disclosure Statement/PTO-1449 ☐ Copies of IDS Citations
12. ☐ Preliminary Amendment
13. ☒ Acknowledgment postcard
14. ☒ Certificate of Mailing
- ☐ First Class ☒ Express Mail *(Specify Label No.):* EL361716631US

UTILITY PATENT APPLICATION TRANSMITTAL (Large Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No.
2204/A12

Total Pages in this Submission
32

Accompanying Application Parts (Continued)

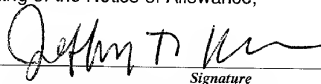
15. ☐ Certified Copy of Priority Document(s) (if foreign priority is claimed)
16. ☐ Additional Enclosures (please identify below):

Fee Calculation and Transmittal

CLAIMS AS FILED

For	#Filed	#Allowed	#Extra		Rate	Fee
Total Claims	13	- 20 =	0	x	\$18.00	\$0.00
Indep. Claims	6	- 3 =	3	x	\$78.00	\$234.00
Multiple Dependent Claims (check if applicable) <input type="checkbox"/>						\$0.00
BASIC FEE						\$690.00
OTHER FEE (specify purpose)						\$0.00
TOTAL FILING FEE						\$924.00

- ☐ A check in the amount of _____ to cover the filing fee is enclosed.
- ☐ The Commissioner is hereby authorized to charge and credit Deposit Account No. _____ as described below. A duplicate copy of this sheet is enclosed.
- ☐ Charge the amount of _____ as filing fee.
 - ☐ Credit any overpayment.
 - ☐ Charge any additional filing fees required under 37 C.F.R. 1.16 and 1.17.
 - ☐ Charge the issue fee set in 37 C.F.R. 1.18 at the mailing of the Notice of Allowance, pursuant to 37 C.F.R. 1.311(b).


Signature

Jeffrey T. Klayman, Reg. No. 39,250
BROMBERG & SUNSTEIN LLP
125 Summer Street
Boston, MA 02110
(617) 443-9292

Dated: February 24, 2000

CC:

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR UNITED STATES PATENT

FOR

**APPARATUS AND METHOD FOR BINDING A CONNECTION-ORIENTED
CLIENT TO A COMMUNICATION CHANNEL**

Inventor:

Richard Crump
295 Beacon Street
Boston, MA 02117

Attorney Docket: 2204/A12 (BA375)

Attorneys:

BROMBERG & SUNSTEIN LLP
125 Summer Street
Boston, MA 02110
(617) 443-9292

-1-

APPARATUS AND METHOD FOR BINDING A CONNECTION-ORIENTED CLIENT TO A COMMUNICATION CHANNEL

PRIORITY

5 The present patent application claims priority from the commonly-owned United States Provisional Patent Application No. 60/121,661 entitled CORRELATING DATA STREAMS OF DIFFERENT PROTOCOLS, which was filed on February 25, 1999 in the names of Richard Crump, Janet Doong, Okoziem Allen, and Debra Guttormsen, and is
10 hereby incorporated herein by reference in its entirety.

CROSS-REFERENCE TO RELATED APPLICATIONS

15 The present patent application may be related to the following commonly-owned United States patent applications, which are hereby incorporated herein by reference in their entireties:

20 U.S. Patent Application No. (2204/169) entitled CORRELATING DATA STREAMS OF DIFFERENT PROTOCOLS, which was filed in the names of Richard Crump and Janet Doong on even date herewith;

U.S. Patent Application No. (2204/A13) entitled MANAGING ADDRESSES IN A COMMUNICATION SYSTEM, which was filed in the names of Richard Crump and Debra Guttormsen on even date herewith; and

25 U.S. Patent Application No. (2204/A14) entitled ENCODING ADDRESSES IN A COMMUNICATION SYSTEM, which was filed in the names of Okoziem Allen and Richard Crump on even date herewith.

FIELD OF THE INVENTION

30 The present invention relates generally to communication systems, and more particularly to binding a connection-oriented client to a communication channel.

-2-

BACKGROUND OF THE INVENTION

In today's information age, communication networks are often used for
5 interconnecting various network devices. Each network device supports various applications that communicate using various protocols. For convenience, an application in a network device is referred to hereinafter as a "client."

In order for a client to communicate over the communication network, the client
10 requires a communication channel. The network device supports various communication channels. The client binds to a communication channel in order to communicate over the communication channel.

In a prior art embodiment of the invention, the network device includes binding
15 logic that binds clients to communication channels. The binding logic is logically situated between the client logic and the X.25 logic within the network device. The binding logic binds all clients having the same destination to a shared set of communication channels. This type of binding does not work well for connection-oriented clients.

SUMMARY OF THE INVENTION

In accordance with one aspect of the invention, each connection-oriented client is
20 bound to its own communication channel.

In one embodiment of the invention, each connection-oriented client is bound to its
25 own X.25 logical channel. The X.25 logical channel includes a single virtual circuit.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects and advantages of the invention will be
30 appreciated more fully from the following further description thereof with reference to the accompanying drawings wherein:

FIG. 1 is a block diagram showing the relationship between the client logic, the
binding logic, and the X.25 logic in an exemplary embodiment of the present invention;

-3-

FIG. 2 is an event flow diagram showing the event flow for initiating a binding by the client logic in an exemplary embodiment of the present invention;

FIG. 3 is a logic flow diagram showing exemplary logic for initiating a binding by the client logic in an exemplary embodiment of the present invention;

FIG. 4 is an event flow diagram showing the event flow for initiating a binding by the X.25 logic in an exemplary embodiment of the present invention; and

FIG. 5 is a logic flow diagram showing exemplary logic for initiating a binding by the X.25 logic in an exemplary embodiment of the present invention.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

In an embodiment of the invention, the network device includes client logic supporting a number of clients, binding logic, and driver logic supporting various types of communication channels. The binding logic is logically situated between the client logic and the driver logic within the network device, and in particular between the client logic and the X.25 logic. The binding logic sets up bindings between clients and communication channels. The binding logic provides a common interface to the clients for all communication channels, and thus the clients do not require any logic that is specific to a particular type of communication channel.

In one embodiment of the invention, the communication channels are X.25 logical channels. X.25 is a common networking protocol that includes a frame-based link layer and a packet-based network layer. X.25 uses virtual circuits in order to multiplex multiple communication streams over a single link layer connection. An X.25 logical channel may include one or more virtual circuits. In such an embodiment, the driver logic includes X.25 logic for managing X.25 logical channels.

FIG. 1 is a block diagram 100 showing the relationship between the client logic 102, the binding logic 104, and the X.25 logic 106 in an exemplary embodiment of the present invention. The binding logic 104 is logically situated between the client logic 102 and the X.25 logic 106. The binding logic sets up bindings between clients and X.25 logical channels. Such bindings may be initiated by the client logic 102, for example,

upon requesting an X.25 communication channel, or by the X.25 logic 106, for example, upon receiving call over the X.25 network.

When a connection-oriented client binds to an X.25 logical channel, the binding logic 104 binds the connection-oriented client to its own X.25 logical channel. The X.25 logical channel includes a single virtual circuit, and is referenced by an X.25 channel identifier. No other clients utilize the communication channel.

In order to simplify the design of the client logic 102, the client logic 102 creates connect logic for each communication channel to which it binds. The connect logic is a separate operating system process from the client logic 102 that is referenced by a connect logic identifier. The connect logic maintains a list of available communication channels, which, for a connection-oriented client, will contain at most one communication channel. Data is exchanged between the client logic 102 and the connect logic using a message passing scheme based upon the connect logic identifier.

The binding logic 104 actually sets up a binding between the connect logic created by the client logic 102 and an X.25 logical channel created by the X.25 logic 106. Data is exchanged between the connect logic and the X.25 logical channel using a message passing scheme based upon the connect logic identifier and the X.25 channel identifier.

INITIATING A BINDING BY THE CLIENT LOGIC

The network device may permit a connection-oriented client 102 to initiate a connection. In this case, when the client 102 requests a communication channel, the communication channel is created, and the client 102 is bound to the communication channel.

When the connection-oriented client logic 102 requires an X.25 logical channel, the client logic 102 creates connect logic for the connection, and sends a channel request message to the binding logic 104. The channel request message includes a calling address, a called address, a connect type indicating that a connection-oriented connection is requested, the connect logic identifier identifying the connect logic, and an X.25 channel identifier equal to zero indicating that a new X.25 logical channel is requested.

Upon receiving the channel request message from the client logic 102, the binding logic 104 sends a channel request message to the X.25 logic 106. The channel request message includes the calling address, the called address, the connect type, and the connect logic identifier.

5 Upon receiving the channel request message from the binding logic 104, the X.25 logic 106 creates an X.25 channel having an X.25 channel identifier, and sends a channel available message to the binding logic 104 indicating that the X.25 channel was created. The channel available message includes the X.25 channel identifier to identify the X.25 channel, and also includes the connect logic identifier so that the binding logic 104 can correlate the X.25 channel with the connect logic.

10 Upon receiving the channel available message from the X.25 logic 106, the binding logic 104 updates its binding database to bind the connect logic identified by the connect logic identifier and the X.25 channel identified by the X.25 channel identifier, and sends a startup message to the connect logic including the X.25 channel identifier. The startup message confirms that the binding was successfully created.

15 Upon receiving the startup message from the binding logic 104, the connect logic updates its list of available communication channels to include the newly created X.25 channel.

20 Thereafter, when the client logic 102 needs to send data over the communication channel, the client logic 102 forwards the data to the connect logic. As part of its normal processing, the connect logic searches its list of available communication channels for a usable channel. Since there is only one communication channel for the connection-oriented client (i.e., the X.25 channel that was created by the X.25 logic 106), the connect logic forwards the data to the X.25 channel.

25 Similarly, when data is received over the X.25 channel, the data is forwarded to the connect logic, which in turn forwards the data to the client logic 102.

30 FIG. 2 shows the various event flows when a binding is initiated by the client logic 102. For convenience, an event flow in which an entity is created is shown by a dashed arrow. An event flow involving a message exchange is shown by a solid arrow. The event sequence number is shown in parentheses.

When the connection-oriented client logic 102 requires an X.25 logical channel, the client logic 102 creates connect logic 108 for the connection in step (1), and sends a channel request message to the binding logic 104 in step (2). The channel request message includes a calling address, a called address, a connect type indicating that a connection-oriented connection is requested, the connect logic identifier identifying the connect logic 108, and an X.25 channel identifier equal to zero indicating that a new X.25 logical channel is requested.

Upon receiving the channel request message from the client logic 102 in step (2), the binding logic 104 sends a channel request message to the X.25 logic 106 in step (3). The channel request message includes the calling address, the called address, the connect type, and the connect logic identifier.

Upon receiving the channel request message from the binding logic 104 in step (3), the X.25 logic 106 creates an X.25 channel 110 having an X.25 channel identifier in step (4), and sends a channel available message to the binding logic 104 indicating that the X.25 channel was created in step (5). The channel available message includes the X.25 channel identifier to identify the X.25 channel 110, and also includes the connect logic identifier so that the binding logic 104 can correlate the X.25 channel 110 with the connect logic 108.

Upon receiving the channel available message from the X.25 logic 106 in step (5), the binding logic 104 updates its binding database to bind the connect logic 108 identified by the connect logic identifier and the X.25 channel 110 identified by the X.25 channel identifier, and sends a startup message to the connect logic 108 including the X.25 channel identifier in step (6). The startup message confirms that the binding was successfully created.

Upon receiving the startup message from the binding logic 104 in step (6), the connect logic 108 updates its list of available communication channels to include the newly created X.25 channel 110.

Thereafter, when the client logic 102 needs to send data over the communication channel, the client logic 102 forwards the data to the connect logic 108 in step (7). As part of its normal processing, the connect logic 108 searches its list of available communication

channels for a usable channel. Since there is only one communication channel for the connection-oriented client (i.e., the X.25 channel 110 that was created by the X.25 logic 106), the connect logic 108 forwards the data to the X.25 channel 110 in step (8).

Similarly, when data is received over the X.25 channel, the data is forwarded to the connect logic, which in turn forwards the data to the client logic 102. This event flow is not shown in FIG. 2.

FIG. 3 is a logic flow diagram showing exemplary logic 300 for initiating a binding by the client logic 102. Beginning at step 302, the client logic 102 creates connect logic having a connect logic identifier, in step 304, and sends a channel request message to the binding logic 104, in step 306. The channel request message includes a calling address, a called address, a connect type indicating that a connection-oriented connection is requested, the connect logic identifier identifying the connect logic, and an X.25 channel identifier equal to zero indicating that a new X.25 logical channel is requested.

Upon receiving the channel request message from the client logic 102, the binding logic 104 sends a channel request message to the X.25 logic 106, in step 308. The channel request message includes the calling address, the called address, the connect type, and the connect logic identifier.

Upon receiving the channel request message from the binding logic 104, the X.25 logic 106 creates an X.25 channel identifier having an X.25 channel identifier, in step 310, and sends a channel available message to the binding logic 104 indicating that the X.25 channel was created, in step 312. The channel available message includes the X.25 channel identifier to identify the X.25 channel, and also includes the connect logic identifier so that the binding logic 104 can correlate the X.25 channel with the connect logic.

Upon receiving the channel available message from the X.25 logic 106, the binding logic 104 updates its binding database to bind the connect logic identified by the connect logic identifier and the X.25 channel identified by the X.25 channel identifier, in step 314, and sends a startup message to the connect logic including the X.25 channel identifier, in step 316. The startup message confirms that the binding was successfully created.

Upon receiving the startup message from the binding logic 104, the connect logic updates its list of available communication channels to include the newly created X.25 channel, in step 318. The logic 300 terminates in step 399.

INITIATING A BINDING BY THE X.25 LOGIC

The network device may be configured to receive calls that are initiated by other network devices. In this case, a call is only accepted if there is a client 102 that is available to be bound to the connection. Thus, a connection-oriented client 102 may make itself available for an incoming call by registering with the X.25 logic 106. The client 102 may register to receive calls from a specific address or from any address.

When the X.25 logic 106 receives a call including a calling address and a called address, the X.25 logic 106 creates an X.25 channel having an X.25 channel identifier. The X.25 logic 106 then sends a new channel message to the client logic 102, and also sends a channel available message to the binding logic 104. The new channel message includes the calling address, the called address, and the X.25 channel identifier. The channel available message includes the X.25 channel identifier, and also includes a connect logic identifier equal to zero to indicate that there is no connect logic associated with the X.25 channel.

Upon receiving the channel available message from the X.25 logic 106, the binding logic 104 updates its binding database to include the newly created X.25 channel.

Upon receiving the new channel message from the X.25 logic 106, the client logic 102 creates connect logic for the connection, and sends a channel request message to the binding logic 104. The channel request message includes the calling address, the called address, a connect type indicating that a connection-oriented connection is requested, the connect logic identifier identifying the connect logic, and the X.25 channel identifier identifying the X.25 channel.

Upon receiving the channel request message from the client logic 102, the binding logic 104 binds the connect logic identified by the connect logic identifier and the X.25 channel identified by the X.25 channel identifier, and sends a startup message to the

connect logic including the X.25 channel identifier. The startup message confirms that the binding was successfully created.

Upon receiving the startup message from the binding logic 104, the connect logic updates its list of available communication channels to include the newly created X.25 channel.

Thereafter, when data is received over the X.25 channel, the data is forwarded to the connect logic, which in turn forwards the data to the client logic 102.

Similarly, when the client logic 102 needs to send data over the communication channel, the client logic 102 forwards the data to the connect logic. As part of its normal processing, the connect logic searches its list of available communication channels for a usable channel. Since there is only one communication channel for the connection-oriented client (i.e., the X.25 channel that was created by the X.25 logic 106), the connect logic forwards the data to the X.25 channel.

FIG. 4 shows the various event flows when a binding is initiated by the X.25 logic 106. For convenience, an event flow in which an entity is created is shown by a dashed arrow. An event flow involving a message exchange is shown by a solid arrow. The event sequence number is shown in parentheses.

When the X.25 logic 106 receives a call including a calling address and a called address, the X.25 logic 106 creates an X.25 channel 110 having an X.25 channel identifier in step (1). The X.25 logic 106 then sends a new channel message to the client logic 102 in step (2), and also sends a channel available message to the binding logic 104 in step (3). The new channel message includes the calling address, the called address, and the X.25 channel identifier. The channel available message includes the X.25 channel identifier, and also includes a connect logic identifier equal to zero to indicate that there is no connect logic associated with the X.25 channel.

Upon receiving the channel available message from the X.25 logic 106 in step (3), the binding logic 104 updates its binding database to include the newly created X.25 channel 110.

Upon receiving the new channel message from the X.25 logic 106 in step (2), the client logic 102 creates connect logic 108 for the connection in step (4), and sends a

-10-

channel request message to the binding logic 104 in step (5). The channel request message includes the calling address, the called address, a connect type indicating that a connection-oriented connection is requested, the connect logic identifier identifying the connect logic 108, and the X.25 channel identifier identifying the X.25 channel 110.

5 Upon receiving the channel request message from the client logic 102 in step (5), the binding logic 104 binds the connect logic 108 identified by the connect logic identifier and the X.25 channel 110 identified by the X.25 channel identifier, and sends a startup message to the connect logic 108 including the X.25 channel identifier in step (6). The startup message confirms that the binding was successfully created.

10 Upon receiving the startup message from the binding logic 104, the connect logic 108 updates its list of available communication channels to include the newly created X.25 channel 110.

15 After the connect logic 108 is bound to the X.25 channel 110, the client logic 102 sends an accept channel message to the X.25 logic in step (7). The accept channel message includes the X.25 channel identifier.

20 Thereafter, when data is received over the X.25 channel 110, the data is forwarded to the connect logic 108 in step (8), which in turn forwards the data to the client logic 102 in step (9).

25 Similarly, when the client logic 102 needs to send data over the communication channel, the client logic 102 forwards the data to the connect logic 108 in step (10). As part of its normal processing, the connect logic 108 searches its list of available communication channels for a usable channel. Since there is only one communication channel for the connection-oriented client (i.e., the X.25 channel that was created by the X.25 logic 106), the connect logic 108 forwards the data to the X.25 channel 110 in step (11).

30 FIG. 5 is a logic flow diagram showing exemplary logic 500 for initiating a binding by the X.25 logic 106. Beginning at step 502, the client logic 102 registers with the X.25 logic 106 as a connection-oriented client available to be bound, in step 504. When the X.25 logic 106 receives a call including a calling address and a called address, in step 506, the X.25 logic 106 creates an X.25 channel having an X.25 channel identifier, in step 508.

-11-

The X.25 logic 106 then sends a new channel message to the client logic 102, in step 510, and also sends a channel available message to the binding logic 104, in step 512. The new channel message includes the calling address, the called address, and the X.25 channel identifier. The channel available message includes the X.25 channel identifier, and also includes a connect logic identifier equal to zero to indicate that there is no connect logic associated with the X.25 channel.

Upon receiving the channel available message from the X.25 logic 106, the binding logic 104 updates its binding database to include the newly created X.25 channel, in step 514.

Upon receiving the new channel message from the X.25 logic 106, the client logic 102 creates connect logic for the connection, in step 516, and sends a channel request message to the binding logic 104, in step 518. The channel request message includes the calling address, the called address, a connect type indicating that a connection-oriented connection is requested, the connect logic identifier identifying the connect logic, and the X.25 channel identifier identifying the X.25 channel.

Upon receiving the channel request message from the client logic 102, the binding logic 104 binds the connect logic identified by the connect logic identifier and the X.25 channel identified by the X.25 channel identifier, and sends a startup message to the connect logic including the X.25 channel identifier, in step 520. The startup message confirms that the binding was successfully created.

Upon receiving the startup message from the binding logic 104, the connect logic updates its list of available communication channels to include the newly created X.25 channel, in step 522. The client logic 102 then sends an accept channel message to the X.25 logic 106 including the X.25 channel identifier, in step 524. The logic 500 terminates in step 599.

ERROR RECOVERY

During the normal course of operation, it is possible for either the client 102 or an X.25 channel to terminate. If a bound client terminates, then the binding logic 104

-12-

performs certain functions to "clean up" its associated X.25 logical channel. If a bound X.25 channel terminates, then the binding logic 104 notifies the client 102 that the communication channel is no longer available. In either case, the binding logic 104 unbinds the client 102 and its associated communication channel.

ALTERNATIVE EMBODIMENTS

In the various embodiments described heretofore, a connection-oriented client was bound to an X.25 communication channel. However, the present invention is in no way limited to binding connection-oriented clients to X.25 communication channels.

An exemplary embodiment of the present invention utilizes binding logic to bind a connection-oriented client to a communication channel. The binding logic provides a common interface to the clients for all communication channels, and is therefore a convenient implementational element of an exemplary embodiment of the invention. However, the present invention does not require binding logic, as similar logic may be included in either the client logic or the driver logic. Thus, the present invention is in no way limited to using binding logic.

Similarly, an exemplary embodiment of the present invention utilizes connect logic to simplify the design of the client, and is therefore a convenient implementational element of an exemplary embodiment of the invention. However, the present invention does not require separate connect logic, as similar logic may be included in the client. Thus, the present invention is in no way limited to using connect logic.

In an exemplary embodiment of the present invention, predominantly all of the logic described herein is implemented as a set of computer program instructions that are stored in a computer readable medium and executed by an embedded microprocessor system within a network device. Various embodiments of the invention may be implemented in any conventional computer programming language. For example, an embodiment may be implemented in a procedural programming language (*e.g.*, "C") or an object oriented programming language (*e.g.*, "C++"). Alternative embodiments of the invention may be implemented using discrete components, integrated circuitry,

-13-

programmable logic used in conjunction with a programmable logic device such as a Field Programmable Gate Array (FPGA) or microprocessor, or any other means including any combination thereof.

Alternative embodiments of the invention may be implemented as a computer program product for use with a computer system. Such implementation may include a series of computer instructions fixed either on a tangible medium, such as a computer readable media (*e.g.*, a diskette, CD-ROM, ROM, or fixed disk), or fixed in a computer data signal embodied in a carrier wave that is transmittable to a computer system via a modem or other interface device, such as a communications adapter connected to a network over a medium. The medium may be either a tangible medium (*e.g.*, optical or analog communications lines) or a medium implemented with wireless techniques (*e.g.*, microwave, infrared or other transmission techniques). The series of computer instructions embodies all or part of the functionality previously described herein with respect to the system. Those skilled in the art should appreciate that such computer instructions can be written in a number of programming languages for use with many computer architectures or operating systems. Furthermore, such instructions may be stored in any memory device, such as semiconductor, magnetic, optical or other memory devices, and may be transmitted using any communications technology, such as optical, infrared, microwave, or other transmission technologies. It is expected that such a computer program product may be distributed as a removable medium with accompanying printed or electronic documentation (*e.g.*, shrink wrapped software), preloaded with a computer system (*e.g.*, on system ROM or fixed disk), or distributed from a server or electronic bulletin board over the network (*e.g.*, the Internet or World Wide Web).

The present invention may be embodied in other specific forms without departing from the essence or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive.

-14-

I claim:

1. A method for binding a connection-oriented client to a communication channel, the method comprising:

5 creating a communication channel for the connection-oriented client, to communication channel having a channel identifier; and

binding the communication channel to the connection-oriented client based upon the channel identifier.

10 2. The method of claim 1, wherein the communication channel is an X.25 logical channel, and wherein the channel identifier is an X.25 channel identifier.

15 3. The method of claim 2, wherein binding the communication channel to the connection-oriented client based upon the channel identifier comprises including the channel identifier in binding messages.

4. The method of claim 1, further comprising forwarding data by the connection-oriented client over the communication channel based upon the channel identifier.

20 5. The method of claim 1, further comprising forwarding data from the communication channel to the connection-oriented client based upon the channel identifier.

-15-

6. A network device comprising:
connection-oriented client logic;
binding logic; and
driver logic, wherein the driver logic is operably coupled to create a
communication channel for the connection-oriented client logic, the communication
channel having a channel identifier, and wherein the binding logic is operably coupled to
bind the connection-oriented client logic and the communication channel using the channel
identifier.

7. The network device of claim 6, wherein the driver logic includes X.25 logic for
creating an X.25 communication channel having an X.25 channel identifier, and wherein
the binding logic binds the connection-oriented client and the X.25 communication
channel using the X.25 channel identifier.

-16-

8. A program product comprising a computer readable medium having embodied therein a computer program for binding a connection-oriented client to a communication channel,

connection-oriented client logic;

binding logic; and

driver logic, wherein the driver logic is programmed to create a communication channel for the connection-oriented client logic, the communication channel having a channel identifier, and wherein the binding logic is programmed to bind the connection-oriented client logic and the communication channel using the channel identifier.

9. The program product of claim 8, wherein the driver logic includes X.25 logic for creating an X.25 communication channel having an X.25 channel identifier, and wherein the binding logic binds the connection-oriented client and the X.25 communication channel using the X.25 channel identifier.

-17-

10. A method comprising:
- requesting a channel by a connection-oriented client;
 - creating the channel by a driver, the channel having a channel identifier; and
 - binding the channel to the connection-oriented client based upon the channel identifier.

5

1100
1099
1098
1097
1096
1095
1094
1093
1092
1091
1090
1089
1088
1087
1086
1085
1084
1083
1082
1081
1080
1079
1078
1077
1076
1075
1074
1073
1072
1071
1070
1069
1068
1067
1066
1065
1064
1063
1062
1061
1060
1059
1058
1057
1056
1055
1054
1053
1052
1051
1050
1049
1048
1047
1046
1045
1044
1043
1042
1041
1040
1039
1038
1037
1036
1035
1034
1033
1032
1031
1030
1029
1028
1027
1026
1025
1024
1023
1022
1021
1020
1019
1018
1017
1016
1015
1014
1013
1012
1011
1010
1009
1008
1007
1006
1005
1004
1003
1002
1001
1000
999
998
997
996
995
994
993
992
991
990
989
988
987
986
985
984
983
982
981
980
979
978
977
976
975
974
973
972
971
970
969
968
967
966
965
964
963
962
961
960
959
958
957
956
955
954
953
952
951
950
949
948
947
946
945
944
943
942
941
940
939
938
937
936
935
934
933
932
931
930
929
928
927
926
925
924
923
922
921
920
919
918
917
916
915
914
913
912
911
910
909
908
907
906
905
904
903
902
901
900
899
898
897
896
895
894
893
892
891
890
889
888
887
886
885
884
883
882
881
880
879
878
877
876
875
874
873
872
871
870
869
868
867
866
865
864
863
862
861
860
859
858
857
856
855
854
853
852
851
850
849
848
847
846
845
844
843
842
841
840
839
838
837
836
835
834
833
832
831
830
829
828
827
826
825
824
823
822
821
820
819
818
817
816
815
814
813
812
811
810
809
808
807
806
805
804
803
802
801
800
799
798
797
796
795
794
793
792
791
790
789
788
787
786
785
784
783
782
781
780
779
778
777
776
775
774
773
772
771
770
769
768
767
766
765
764
763
762
761
760
759
758
757
756
755
754
753
752
751
750
749
748
747
746
745
744
743
742
741
740
739
738
737
736
735
734
733
732
731
730
729
728
727
726
725
724
723
722
721
720
719
718
717
716
715
714
713
712
711
710
709
708
707
706
705
704
703
702
701
700
699
698
697
696
695
694
693
692
691
690
689
688
687
686
685
684
683
682
681
680
679
678
677
676
675
674
673
672
671
670
669
668
667
666
665
664
663
662
661
660
659
658
657
656
655
654
653
652
651
650
649
648
647
646
645
644
643
642
641
640
639
638
637
636
635
634
633
632
631
630
629
628
627
626
625
624
623
622
621
620
619
618
617
616
615
614
613
612
611
610
609
608
607
606
605
604
603
602
601
600
599
598
597
596
595
594
593
592
591
590
589
588
587
586
585
584
583
582
581
580
579
578
577
576
575
574
573
572
571
570
569
568
567
566
565
564
563
562
561
560
559
558
557
556
555
554
553
552
551
550
549
548
547
546
545
544
543
542
541
540
539
538
537
536
535
534
533
532
531
530
529
528
527
526
525
524
523
522
521
520
519
518
517
516
515
514
513
512
511
510
509
508
507
506
505
504
503
502
501
500
499
498
497
496
495
494
493
492
491
490
489
488
487
486
485
484
483
482
481
480
479
478
477
476
475
474
473
472
471
470
469
468
467
466
465
464
463
462
461
460
459
458
457
456
455
454
453
452
451
450
449
448
447
446
445
444
443
442
441
440
439
438
437
436
435
434
433
432
431
430
429
428
427
426
425
424
423
422
421
420
419
418
417
416
415
414
413
412
411
410
409
408
407
406
405
404
403
402
401
400
399
398
397
396
395
394
393
392
391
390
389
388
387
386
385
384
383
382
381
380
379
378
377
376
375
374
373
372
371
370
369
368
367
366
365
364
363
362
361
360
359
358
357
356
355
354
353
352
351
350
349
348
347
346
345
344
343
342
341
340
339
338
337
336
335
334
333
332
331
330
329
328
327
326
325
324
323
322
321
320
319
318
317
316
315
314
313
312
311
310
309
308
307
306
305
304
303
302
301
300
299
298
297
296
295
294
293
292
291
290
289
288
287
286
285
284
283
282
281
280
279
278
277
276
275
274
273
272
271
270
269
268
267
266
265
264
263
262
261
260
259
258
257
256
255
254
253
252
251
250
249
248
247
246
245
244
243
242
241
240
239
238
237
236
235
234
233
232
231
230
229
228
227
226
225
224
223
222
221
220
219
218
217
216
215
214
213
212
211
210
209
208
207
206
205
204
203
202
201
200
199
198
197
196
195
194
193
192
191
190
189
188
187
186
185
184
183
182
181
180
179
178
177
176
175
174
173
172
171
170
169
168
167
166
165
164
163
162
161
160
159
158
157
156
155
154
153
152
151
150
149
148
147
146
145
144
143
142
141
140
139
138
137
136
135
134
133
132
131
130
129
128
127
126
125
124
123
122
121
120
119
118
117
116
115
114
113
112
111
110
109
108
107
106
105
104
103
102
101
100
99
98
97
96
95
94
93
92
91
90
89
88
87
86
85
84
83
82
81
80
79
78
77
76
75
74
73
72
71
70
69
68
67
66
65
64
63
62
61
60
59
58
57
56
55
54
53
52
51
50
49
48
47
46
45
44
43
42
41
40
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0

-18-

11. A method comprising:

registering to receive a call by a connection-oriented client;

receiving a call by a driver;

creating a channel by the driver, the channel having a channel identifier; and

5 binding the channel to the connection-oriented client based upon the channel identifier.

11. A method comprising:
registering to receive a call by a connection-oriented client;
receiving a call by a driver;
creating a channel by the driver, the channel having a channel identifier; and
binding the channel to the connection-oriented client based upon the channel identifier.

12. A binding message for binding a connection-oriented client and a communication channel, the binding message comprising a channel identifier for the communication channel.

13. The binding message of claim 12, wherein the communication channel is an X.25 logical channel, and wherein the channel identifier is an X.25 channel identifier.

ABSTRACT OF THE DISCLOSURE

Each connection-oriented client in a network device is bound to its own communication channel. The communication channel may be an X.25 logical channel having a single virtual circuit.

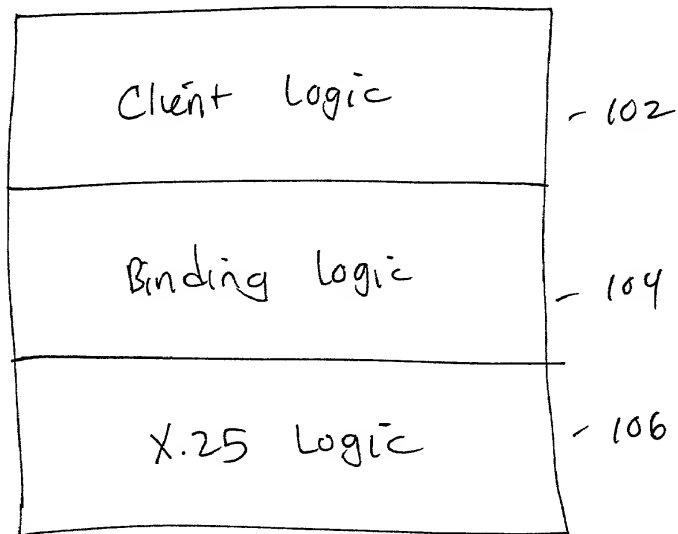


FIG. 1 100

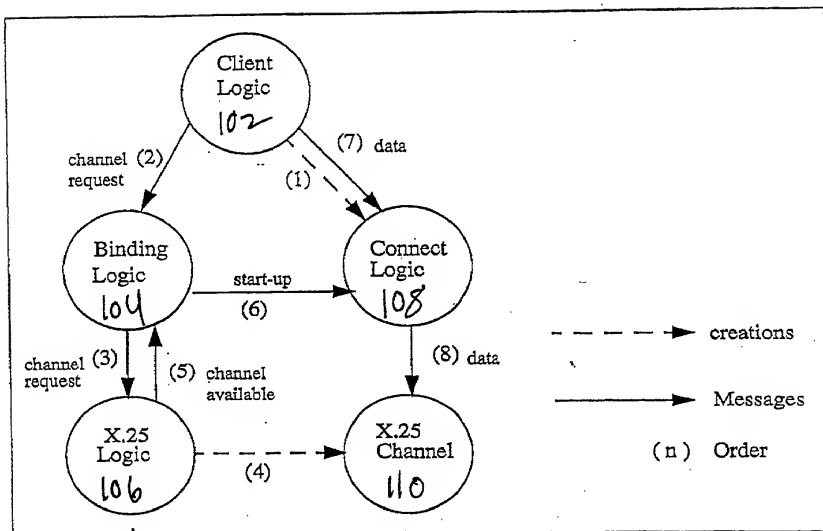


fig. 2 200

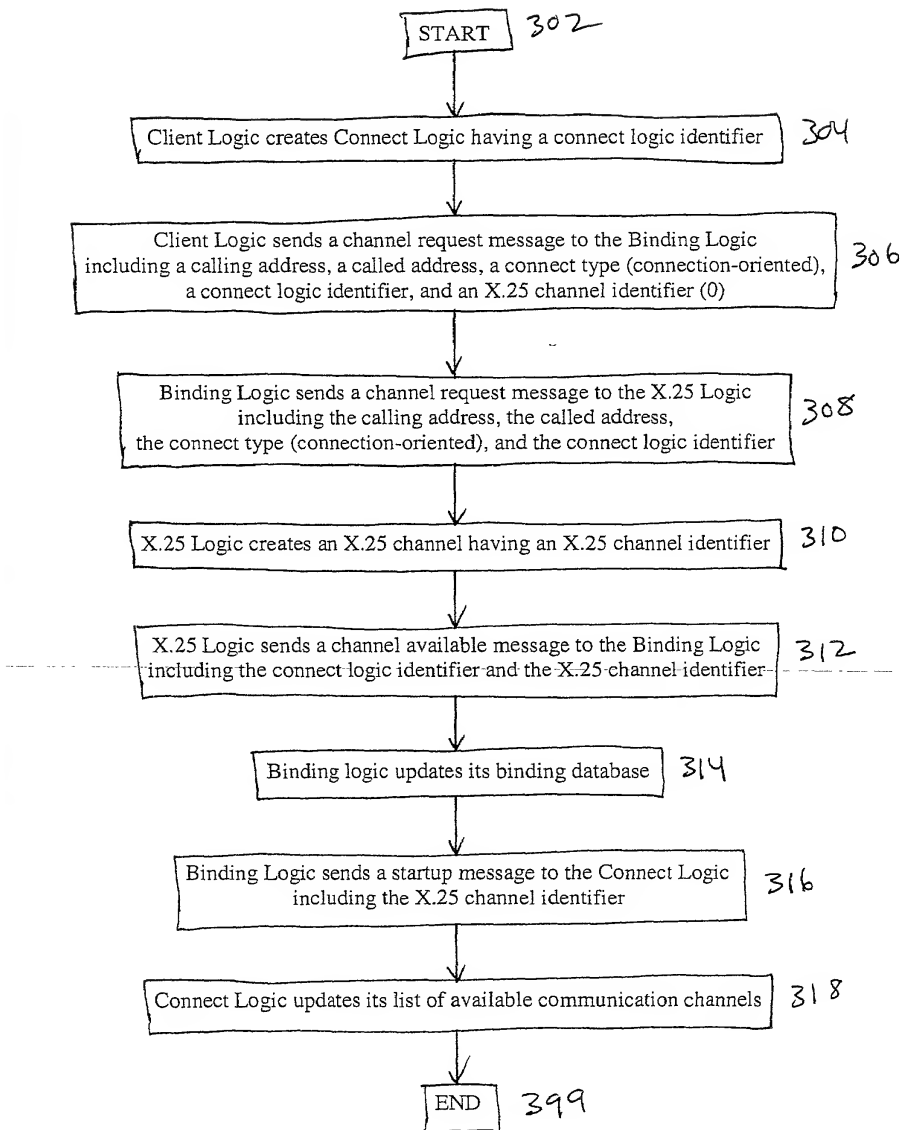


FIG. 3 300

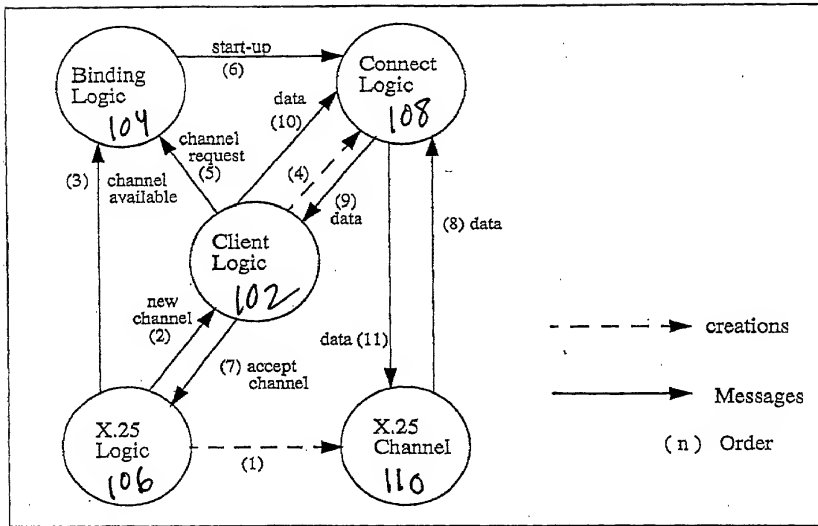


FIG. 4

400

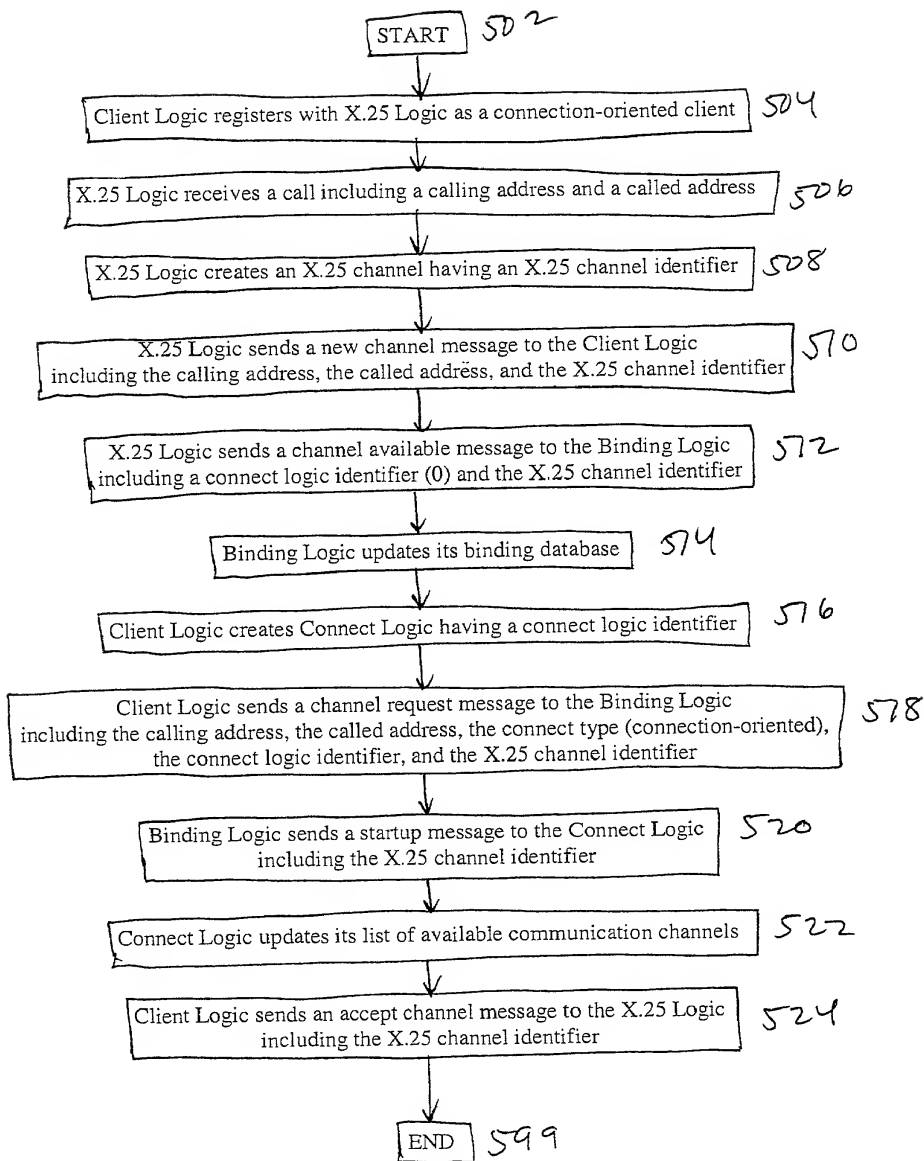


FIG. 5 500

Docket No.
2204/A12

Declaration and Power of Attorney For Patent Application

English Language Declaration

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

APPARATUS AND METHOD FOR BINDING A CONNECTION-ORIENTED CLIENT TO A COMMUNICATION CHANNEL

the specification of which

(check one)

☒ is attached hereto.

☐ was filed on _____ as United States Application No. or PCT International Application Number _____ and was amended on _____ (if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d) or Section 365(b) of any foreign application(s) for patent or inventor's certificate, or Section 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate or PCT International application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)

Priority Not Claimed

_____ (Number)	_____ (Country)	_____ (Day/Month/Year Filed)	<input type="checkbox"/>
_____ (Number)	_____ (Country)	_____ (Day/Month/Year Filed)	<input type="checkbox"/>
_____ (Number)	_____ (Country)	_____ (Day/Month/Year Filed)	<input type="checkbox"/>

I hereby claim the benefit under 35 U.S.C. Section 119(e) of any United States provisional application(s) listed below:

<u>60/121,661</u>	<u>February 25, 1999</u>
(Application Serial No.)	(Filing Date)
<u> </u>	<u> </u>
(Application Serial No.)	(Filing Date)
<u> </u>	<u> </u>
(Application Serial No.)	(Filing Date)

I hereby claim the benefit under 35 U. S. C. Section 120 of any United States application(s), or Section 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 U.S.C. Section 112, I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37, C. F. R., Section 1.56 which became available between the filing date of the prior application and the national or PCT International filing date of this application:

<u> </u>	<u> </u>	<u> </u>
(Application Serial No.)	(Filing Date)	(Status)
		(patented, pending, abandoned)
<u> </u>	<u> </u>	<u> </u>
(Application Serial No.)	(Filing Date)	(Status)
		(patented, pending, abandoned)
<u> </u>	<u> </u>	<u> </u>
(Application Serial No.)	(Filing Date)	(Status)
		(patented, pending, abandoned)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. *(list name and registration number)*

Bruce D. Sunstein	Reg. No. 27,234	Jay Sandvos	Reg. No. 43,900
Robert M. Asher	Reg. No. 30,445	Sonia K. Guterman	Reg. No. 44,729
Timothy M. Murphy	Reg. No. 33,198	Keith J. Wood	Reg. No. 45,235
Steven G. Saunders	Reg. No. 36,265	Mary M. Steubing	Reg. No. 37,946
Harriet M. Strimpel	Reg. No. 37,008	Christopher J. Cianciolo	Reg. No. 42,417
Samuel J. Petuchowski	Reg. No. 37,910	Lindsay J. McGuinness	Reg. No. 38,549
Jeffrey T. Klayman	Reg. No. 39,250		
John J. Stickevers	Reg. No. 39,387		
Herbert A. Newborn	Reg. No. 42,031		
Elizabeth P. Morano	Reg. No. 42,904		
Jean M. Tibbetts	Reg. No. 43,193		

Send Correspondence to: Jeffrey T. Klayman
Bromberg & Sunstein LLP
125 Summer Street
Boston, MA 02110

Direct Telephone Calls to: *(name and telephone number)*
Jeffrey T. Klayman (617) 443-9292

Full name of sole or first inventor Richard Crump	
Sole or first inventor's signature	Date
Residence 295 Beacon Street, Boston, MA 02117	
Citizenship U.S.	
Post Office Address same as residence	

Full name of second inventor, if any	
Second inventor's signature	Date
Residence	
Citizenship	
Post Office Address	